Empower Nested Boolean Logic via Self-Supervised Curriculum Learning

Hongqiu Wu[1,2] and **Linfeng Liu**[1,2] and **Hai Zhao**[1,2*] and **Min Zhang**[3,4]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]School of Computer Science and Technology, Soochow University, Suzhou, China
[4]Harbin Institute of Technology, Shenzhen, China
{wuhongqiu,linfengliu}@sjtu.edu.cn,zhaohai@cs.sjtu.edu.cn,
minzhang@suda.edu.cn

Abstract

Beyond the great cognitive powers showcased by language models, it is crucial to scrutinize whether their reasoning capabilities stem from strong generalization or merely exposure to relevant data. As opposed to constructing increasingly complex logic, this paper probes into the boolean logic, the root capability of a logical reasoner. We find that any pre-trained language models even including large language models only behave like a random selector in the face of multi-nested boolean logic, a task that humans can handle with ease. To empower language models with this fundamental capability, this paper proposes a new self-supervised learning method *Curriculum Logical Reasoning* (CLR), where we augment the training data with nested boolean logic chain step-by-step, and program the training from simpler logical patterns gradually to harder ones. This new training paradigm allows language models to effectively generalize to much harder and longer-hop logic, which can hardly be learned through naive training. Furthermore, we show that boolean logic is a great foundation for improving the subsequent general logical tasks[1].

1 Introduction

Artificial intelligence has made a giant leap from perception to cognition, with powerful pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020; Clark et al., 2020; Raffel et al., 2020; Brown et al., 2020; He et al., 2021b), large language models (LLMs) (Chung et al., 2022; Chowdhery et al., 2022; OpenAI, 2023) demonstrating human-level comprehension and reasoning powers on a series of challenging tasks like commonsense reasoning (Zellers et al., 2019), open-domain question-answering (Mihaylov et al., 2018), arithmetical reasoning (Ling et al., 2017).

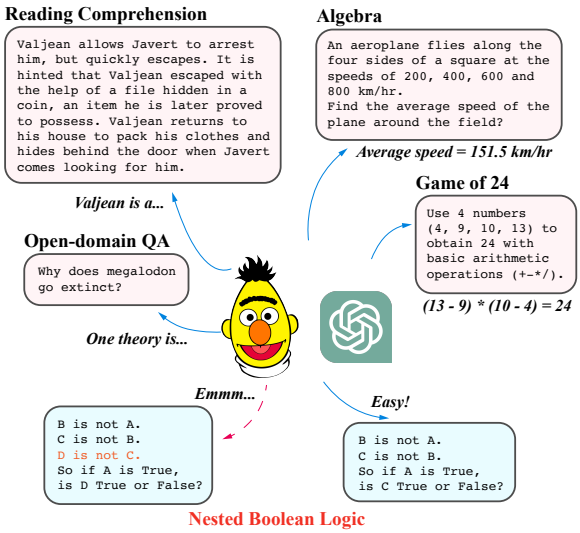


Figure 1: While language models are capable of handling a range of complex logical tasks, they do not perform well on more basic nested boolean logic.

While this is charming, these over-parameterized language models are shown to be good at exploiting superficial statistical cues to achieve decent scores on end tasks (Zhou et al., 2021; Sanyal et al., 2022a; Wu et al., 2023b). Early on BERT, it is found that simply by adding a "not" to the claims, BERT would be fooled into a random selector (Niven and Kao, 2019). It is time to go back and scrutinize whether the state-of-the-art PLMs master solid logical capability, as truly powerful logical reasoners.

Rather than creating even more complex logic, this paper concentrates on the root level of logical reasoning - boolean logic, as in Figure 1. Any logic can be reduced to a combination of multiple boolean operations, including negation $\neg$, intersection $\wedge$, and union $\vee$. In this paper, we introduce a new probing method to quantify the boolean logical reasoning of a language model, fine-grained to different levels of logical difficulty.

However, our results show that none of PLMs

*Corresponding author; This paper was partially supported by Joint Research Project of Yangtze River Delta Science and Technology Innovation Community (No. 2022CSJGG1400).

[1]https://github.com/gingasan/boolkill

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 13731–13742
December 6-10, 2023

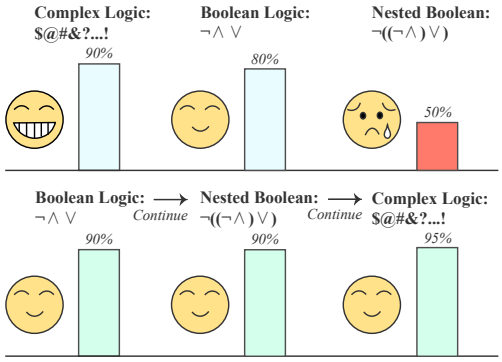


Figure 2: Overview of *Curriculum Logical Reasoning*.

possess the necessary proficiency to tackle the multiple nesting of (multi-nested) simple boolean operations, even the state-of-the-art models like DeBERTa-V3 (He et al., 2021a) and ChatGPT (OpenAI, 2023). Faced with more than three nested boolean operations, they quickly degenerate into a random selector. even with the chain-of-thought prompt (Wei et al., 2022; Zhang et al., 2022b). Conversely, this task is very simple for humans, compared to other more general reasoning tasks. **This raises a shadow over their generalizability acquired from large amount of training.**

To empower the language models with such a fundamental capability in nested boolean logic, we propose a new self-supervised training paradigm, *Curriculum Logical Reasoning* (CLR), inspired by curriculum learning (Bengio et al., 2009). Concretely, we construct the nested boolean logic step-by-step from simple to hard on top of the original training samples in a self-supervised manner (Devlin et al., 2019). The model is encouraged to start with learning simple logical patterns and then move forward to hard ones gradually, rather than learning hard logic with a single leap. We find that recalling simpler logic while learning harder logic can result in a better outcome. Our experiments demonstrate that CLR significantly enhances the logical learning process. Excitingly, pre-learning boolean logic acts as a great foundation step to further enhance the subsequent logical end tasks, like ReClor and DREAM. Figure 2 illustrates CLR very lively.

2 Introducing Nested Boolean Logic

This section presents our method to introduce multi-nested boolean logic to existing data.

We first present the notations. Let $\mathbf{x}$ denote the input text, with its ground truth y, and p_θ denote the classifier (e.g. a language model) with parameters θ. Given an arbitrary input sample $\mathbf{x}$, suppose that the model accurately predicts $p_\theta(\mathbf{x}) = y$. We now define an operation δ on $\mathbf{x}$, which can be regarded as a transformation on the text, denoted as $\delta \cdot \mathbf{x}$.

The earth is flat.	*Original sample*
S_0: The earth is flat. Is S_0 true or false?	*Convert to context-question*
S_0: The earth is flat. S_1: S_0 is a false statement. S_2: S_1 is a false statement. S_3: S_2 is a true statement. Is S_3 true or false?	*Add nested boolean logic* *NOT only*
S_0: The earth is flat. S_1: S_0 is a false statement. S_2: S_1 is a false statement. S_3: Either S_2 or S_1 is a true statement. / S_3: Both S_2 and S_1 are true statements. Is S_3 true or false?	*Add nested boolean logic* *NOT & AND & OR*

Table 1: Method to augment arbitrary samples with nested boolean logic.

2.1 From Simple Boolean Logic to Nested Boolean Logic

We concentrate on the logical operation, which specifically manipulates the underlying logical chain by transformation on the text. We present a new form of logical operation that corresponds to only boolean operators, i.e. intersection $\wedge$, union $\vee$, and negation $\neg$. We might concentrate on the simplest negation first.

Suppose that the input statement $\mathbf{x}$ entails a fact f, which can be either a true fact or a false fact, represented by y_0. The logical process can be formulated as $\mathbf{x} \Rightarrow y_0$, where $\Rightarrow$ refers to "implies that" and $y_0 \in \{0, 1\}$ (0 for *True* and 1 for *False*).

We illustrate a toy example of our logical operation in Table 1. First, the model is required to discriminate whether the stated fact in $\mathbf{x}$ is true or false. It states a false fact "the earth is flat", so $y_0 = 1$ (*False*). Next, we transfer it to a context-question template and denote the context as S_0. It is still a binary classification and the answer for it is limited in *True* or *False*. This template can be applied to arbitrary tasks. For instance, a sentiment analysis sentence "cold movie" can be rewritten to a statement like "cold movie expresses a positive movie watching".

Our idea is to craft a series of statements after S_0. Each statement asserts the truth or falsity of the previous statement, which is uniformly chosen. We denote such a statement as *boolean statement*, and

ask the model to discriminate the final statement. For instance, $y_0 = 1$ and S_1 asserts S_0 is false, so y_1 should be negated, $y_1 = 0$. After deduction, we can obtain $y_3 = 1$.

Logically, the assertion of "true" results in no change of the current logic and the assertion of "false" results in a negation. δ can be nested for k times without affecting the fact in $\mathbf{x}$:

$$\prod_{i=1}^{k} \delta_i \cdot \mathbf{x} \Rightarrow y_k \tag{1}$$

where y_i denotes each intermediate answer after i times of boolean statements and y_k denotes the eventual answer. We denote Eq. 1 as *multi-nested boolean logic*.

Obtaining final y_k is free of external annotation, as in self-supervised learning, by programming the following recursion:

$$y_i = \begin{cases} \neg \quad y_{i-1} & , \delta_i \text{ asserts false} \\ \quad\; y_{i-1} & , \delta_i \text{ asserts true} \end{cases} . \tag{2}$$

Such multi-nested boolean logic poses little challenge to humans. We hopefully assume that a strong language model can tackle that as well.

We generalize the negation operation to other boolean operations as in the bottom of Table 1. Concretely, we uniformly choose one statement from S_1 to S_k and append it with either "and" or "or" chosen uniformly.

2.2 Quantify Boolean Logic

We probe the mastery in nested boolean logic of a language model by measuring its performance against our boolean statements. An ideal logical reasoner is supposed to make clear logical transitions between truth and falsity. We are particularly interested in this situation: **the model accurately discriminates the original fact, while falters in delivering the correct answer subsequent to k boolean statements.** This can be formulated as:

$$p_\theta \left(\prod_{i=1}^{k} \delta_i \cdot \mathbf{x} \right) \neq y_k \tag{3}$$

where p_θ satisfies:

$$p_\theta \left(\mathbf{x} \right) = y_0. \tag{4}$$

Deep neural models are good at exploiting superficial features rather than delving into the entire semantics (Wu et al., 2023a; Sanyal et al., 2022a). The consequence is that they can get the final result without correctly classifying the original fact. Eq. 3 and 4 exclude this potential threat and focus entirely on the model's capability in handling nested boolean logic. In other words, if the model reasons from a misclassified fact, its final result can be noisy, misleading the analysis.

Hence, we are interested in two metrics:

• *Clean accuracy (clean%)*: It refers to the general accuracy score.

• *Boolean accuracy (boolean%)*: It refers to the accuracy only calculated on those samples where the model accurately discriminates the original fact, as represented in Eq. 3 and 4. This can only be calculated on augmented data.

3 Benchmark

To benchmark the multi-nested boolean logic, we construct a new dataset in this paper and following experiments are based on this. As apart from other datasets, it is composed of a series of subsets, representing different levels of logical complexity. We will release this benchmark for future research.

3.1 Data Collection

We collect the raw data from SciTail (Khot et al., 2018), a scientific text entailment dataset with a premise and a hypothesis for each sample, which is labeled as *entail* or *not entail*. We join the premise and hypothesis together to make them a "fact", with the entailed pair labeled as *True* and not entailed one labeled as *False*. Some samples are shown in Appendix A. Eventually, we get 6,000 raw samples and randomly sample 1,000 of them as the test set with the rest as the training set.

On top of the raw data, we convert it to the context-question format and then impose boolean statements to generate the adversarial set, which means that the resultant samples are likely to fool the model (Zellers et al., 2018, 2019). Specifically, we uniformly choose a value k from some range and insert k boolean statements following the original sample. The range of k bounds the minimal and maximal nesting of boolean logic on each sample, and larger value of k suggests more nesting on the logic chain. For instance, the samples in Table 1 correspond to $k = 0$ and $k = 3$ (see Appendix A).

We denote this benchmark as *BoolKill*, in which each sample is a logic chain started with a potential fact and followed by a series of boolean statements. It is worth noting that BoolKill is a group of sets

	DeBERTa-base	DeBERTa-large	GPT2-1.5b
raw	96.4	98.1	96.0
u_0	96.7	97.8	96.8

Table 2: Performances on raw data and its templated u_0.

for different levels of logical difficulty, and each level has its own training and test set. We use the following notations to spot them:

• raw: the raw data in which each sample is a statement of a fact;

• u_0: the clean set in which each raw sample is only transferred to a context-question template, with semantics unchanged;

• u_k: the adversarial set constructed on top of u_0 in which each sample is suffixed by k boolean statements;

• $u_{k_1 \sim k_2}$: the adversarial set in which each sample is suffixed by $k_1 \sim k_2$ boolean statements;

• $\tilde{u}_k / \tilde{u}_{k_1 \sim k_2}$: u is negation-only, and we use $\tilde{u}$ to distinguish the adversarial set additionally containing `AND` and `OR`.

3.2 Data Bias

The first thing to verify is whether u_0 is semantically equivalent to raw. From Table 2, we find that each model achieves very close performances on raw and u_0, suggesting that the context-question template does not induce bias to the original data.

The average sentence length will vary due to the boolean statements on raw data, which grows linearly from 36 to 88, from u_1 to $\tilde{u}_8$. The overall statistics of BoolKill are in Appendix A.

To minimize the bias between subsets, we keep the ratio of positive and negative samples to 1:1 in all subsets. Additionally, BoolKill is a semi-annotated dataset, comprising human-annotated facts and synthetic boolean statements. The latter introduces several high-frequency words like "true", "false", and "statement", which may induce large bias if these words do not occur in balance in data. For instance, the model may make the decision based on the relative number of "true" and "false" in the sentence. Hence, we also keep the occurrence of "true" and "false" the same for both positive and negative samples in all subsets.

3.3 Evaluation Results

We report the thorough results on each level of logical difficulty on BoolKill. We sequentially evaluate each model on u_0, u_1, u_2, ..., and u_8 ($\tilde{u}_8$), indicating the number of nested boolean operations.

We evaluate two state-of-the-art PLMs:

• DeBERTa-V3 (He et al., 2021a): one of the strongest BERT-style language models;

• ChatGPT (OpenAI, 2023): the strongest large language model, as a powerful zero-shot learner.

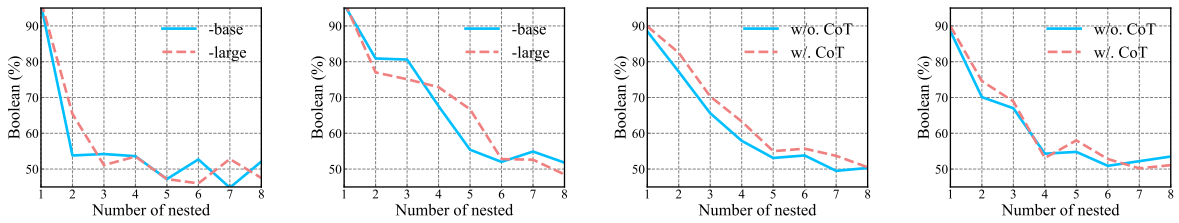
ChatGPT shows an impressive ability to follow human instructions and we directly evaluate it on the test sets[2]. For DeBERTa, we first fine-tune it on the u_k training set and evaluate it on the u_k test.

NOT: We curve the results in Figure 3. We find that each model exhibits a high performance on u_1, suggesting their proficiency in tackling single boolean logic. DeBERTa performs better than ChatGPT, probably due to task-specific fine-tuning. However, as the nesting increases, each model suffers from a notable decline regardless of size. For instance in (a), starting from u_2, in which the samples are suffixed by only two boolean statements, DeBERTa-base falls to 53.8% while DeBERTa-large falls to 65.4%. From u_3, strong as DeBERTa-large, it leans to a random selector, whose accuracy gets close to 50%. Similar situations can be seen on ChatGPT, while its degradation is more gentle. It suggests that even state-of-the-art models possess a critical limitation in the basic nested boolean logic, only able to handle up to three nested operations. This is far below humans' level.

AND & OR: From (b), it is counter-intuitive that DeBERTa performs better on sets additionally including `AND` and `OR`. We conjecture that the model utilizes the inherent bias that `AND` $\Rightarrow$ *False* and `OR` $\Rightarrow$ *True* in majority of cases. Such a shortcut is particularly useful when k is small. Interestingly from (d), well-trained ChatGPT appears not to use this, and its performance drops even faster on $\tilde{u}$. Therefore, we focus on $\tilde{u}$ and $\tilde{u}$ with large k in the following experiments.

Chain-of-Thought (CoT) (Wei et al., 2022; Zhang et al., 2022b; Yao et al., 2023) is proven to be an effective prompt method to amplify the reasoning ability of LLMs, with asking them to offer the procedure while performing the reasoning. From Figure 3 (c) and (d), we find that ChatGPT performs better with the assistance of CoT. However, we raise a criticism in the paper: does CoT promote logical reasoning? Indeed, our study show that CoT may bring new logical concern. We will

[2]We use the API from *openai*. The backbone model is *gpt-3.5-turbo*.



(a) DeBERTa-V3 (u_k) (b) DeBERTa-V3 ($\tilde{u}_k$) (c) ChatGPT (u_k) (d) ChatGPT ($\tilde{u}_k$)

Figure 3: Boolean accuracy of different models with increasing numbers of nested boolean operations (u_k/$\tilde{u}_k$).

further discuss it in Sec. 6.1.

4 Empower Nested Boolean Logic

We present a new self-supervised learning manner.

4.1 Self-Supervised Learning

The straightforward method is to fine-tune the model on BoolKill. The concept behind is to sequentially introduce boolean statements on top of some corpus and let the model learn to tackle multi-nested boolean logic self-supervisedly.

However, we find language models struggle to fit the samples in BoolKill when the potential logic within the data is too hard, and still be a random selector. It indicates that naive training is not the best therapy to learn complex logical patterns.

4.2 Curriculum Logical Reasoning

Inspired by Curriculum Learning (Bengio et al., 2009), where the machine learning model is encouraged to learn the task starting with easier samples and ending with harder ones, we propose *Curriculum Logical Reasoning* (CLR) to enhance the process of learning logical reasoning.

There is a natural match between curriculum learning and logical philosophy, because the logic chain is a step-by-step progression from single to complex. CLR means that, rather than learning hard logic from scratch, the model starts with learning simpler logic, e.g. single boolean logic, and then moves forward to harder logic gradually, e.g. multi-nested boolean logic.

We show a concrete instance. We start to train the model on $u_{0\sim1}$, which solely includes single boolean operations. Next, we train such a model on $u_{0\sim2}$, which further includes two-nested boolean operations. This gradual progression continues until the model is trained on $u_{0\sim4}$. The above procedure can be denoted as $u_{0\sim1} \rightarrow u_{0\sim2} \rightarrow u_{0\sim3} \rightarrow u_{0\sim4}$. We find that reusing the easier samples in the new turn of training benefits the eventual performance, which potentially reminds the model of what it learns previously. Our ultimate goal is that the model can gradually learn to tackle more complex logic that it has not seen before.

5 Empirical Results

As opposed to the prior section, where we evaluate the model on each level of logical difficulty, in this section, we evaluate each model on BoolKill $u_{1\sim4}$, $u_{5\sim8}$, and $\tilde{u}_{5\sim8}$ as an alternative. These sets cover the range from $k = 1$ to $k = 8$. $u_{1\sim4}$ is a simpler one and $u_{5\sim8}$ and $\tilde{u}_{5\sim8}$ appear to be highly challenging, since we previously show that state-of-the-art PLMs are almost powerless for the nested boolean logic beyond u_3.

We experiment on DeBERTa-V3-base and DeBERTa-V3-large. Each model is trained for 3,000 steps with a batch size of 16 and learning rate of 2e-5 / 1e-5 for the base / large one.

To verify CLR, we report two experiments. In the first experiment, we compare different training settings and evaluate the models on BoolKill. In the second, we leverage the boolean logic in BoolKill to benefit other general logical tasks.

5.1 Nested Boolean Logic

The results across various BoolKill sets are summarized in Table 3. We find that naively training the model on $u_{5\sim8}$ only produces random accuracy scores on all three test sets, even on two simpler ones u_0 and $u_{1\sim4}$. While on $u_{0\sim4}$ and $\tilde{u}_{0\sim4}$, we find that DeBERTa-V3-large can achieve better outcomes on simpler $u_{1\sim4}$ compared to DeBERTa-V3-base. It suggests that a larger model possibly has a greater learning ability to handle more nested boolean operations, but it is still very hard even for strong DeBERTa-V3-large, to learn very difficult logical patterns in $u_{5\sim8}$ within a single leap.

However, CLR brings significant performance

		u_0	$u_{1\sim4}$	$u_{5\sim8}$	$\tilde{u}_{5\sim8}$
		clean%	*boolean%*		
					DeBERTa-V3-base
NAIVE	$u_{5\sim8}$	50.2	46.6	49.6	51.2
	$u_{0\sim4}$	96.0	71.5	53.6	53.4
	$\tilde{u}_{0\sim4}$	94.6	72.2	52.9	56.0
CLR	$u_{0\sim1}$	96.2	64.6	49.0	50.5
	$\rightarrow u_{0\sim2}$	**96.4**	89.6 ↑	57.6 ↑	56.7 ↑
	$\rightarrow u_{0\sim3}$	96.1	94.6 ↑	70.0 ↑	60.7 ↑
	$\rightarrow u_{0\sim4}$	96.3	96.8 ↑	**79.2** ↑	66.8 ↑
	$\rightarrow \tilde{u}_{0\sim4}$	95.8	**97.4** ↑	77.5	**73.0** ↑
					DeBERTa-V3-large
NAIVE	$u_{5\sim8}$	55.2	48.6	51.3	50.8
	$u_{0\sim4}$	96.4	97.9	61.9	54.7
	$\tilde{u}_{0\sim4}$	96.1	77.6	51.7	60.8
CLR	$u_{0\sim1}$	97.7	70.2	52.7	48.7
	$\rightarrow u_{0\sim2}$	**98.0**	87.0 ↑	60.7 ↑	55.2 ↑
	$\rightarrow u_{0\sim3}$	97.7	98.5 ↑	71.1 ↑	59.3 ↑
	$\rightarrow u_{0\sim4}$	97.6	99.4 ↑	**84.3** ↑	68.2 ↑
	$\rightarrow \tilde{u}_{0\sim4}$	97.3	**99.5** ↑	81.9	**82.3** ↑

Table 3: Results on BoolKill, comparing CLR with naive training. We use "→" to denote the curriculum setting we perform, where the model inherits the trained weights from the last level. We highlight the step-by-step performance gains CLR brings with "↑".

boosts on every model and every test set. Its advantages are especially significant on harder $u_{5\sim8}$ and $\tilde{u}_{5\sim8}$. For instance on DeBERTa-V3-large, it achieves an impressive boolean accuracy of 84.3% on $u_{5\sim8}$ and 82.3% on $\tilde{u}_{5\sim8}$, uplifting naive training by about 30%, also keeping a high clean accuracy of 97.6% and 97.3% on u_0. It is worth noting that the model has not ever seen the hard samples in $u_{5\sim8}$ and $\tilde{u}_{5\sim8}$ during training, and CLR effectively generalizes the model to unseen logical patterns. Additionally, all models consistently maintain a strong accuracy on u_0 throughout the process of CLR, suggesting that they learn to discriminate the original facts and tackle boolean logic simultaneously. As a contrast, naive self-supervised training leads to inferior u_0 results.

Moreover, we find that each level of curriculum brings a considerable improvement to the model. For instance, the performance of DeBERTa-V3-base has outperformed all naive baselines when it just completes the second level of training on $u_{0\sim2}$.

5.2 Boolean Benefits Complex Logic

Boolean logic acts as the atomic component of logic. Our intuition is that it can solidify more general end tasks that require complex logical reasoning. We conduct validation on two machine reading comprehension (MRC) datasets: • ReClor (Yu et al., 2020), a reasoning-required MRC collected from graduate admission exams; • DREAM (Sun et al., 2019), a dialogue-based MRC. Concretely, we first train DeBERTa-V3 on BoolKill as an initialization and then fine-tune it on the task-specific data of ReClor and DREAM.

		ReClor	**DREAM**
DeBERTa -V3-base	sp	58.2	79.9
	$u_0 \rightarrow sp$	59.0	80.2
	$u_{0\sim1} \rightarrow sp$	61.6 $_{\uparrow 3.4}$	82.0 $_{\uparrow 2.1}$
	$u_{0\sim1} \rightarrow u_{0\sim2} \rightarrow sp$	**62.6** $_{\uparrow 4.4}$	**82.8** $_{\uparrow 2.9}$
DeBERTa -V3-large	sp	71.4	90.4
	$u_{0\sim1} \rightarrow u_{0\sim2} \rightarrow sp$	**74.8** $_{\uparrow 3.4}$	**92.5** $_{\uparrow 2.1}$
LLaMA2 -7b (LoRA)	sp	55.4	85.1
	$u_{0\sim1} \rightarrow u_{0\sim2} \rightarrow sp$	**61.6** $_{\uparrow 6.2}$	**86.9** $_{\uparrow 1.8}$
		$u_{0\sim1}$	
DeBERTa -V3-base	$u_{0\sim1}$	96.6	98.1
	$sp \rightarrow u_{0\sim1}$	95.3 $_{\downarrow 1.3}$	97.8 $_{\downarrow 0.3}$

Table 4: Results on general MRC tasks. "sp" refers to the task-specific training set and we evaluate the model on the corresponding test set.

The results are shown in Table 4. We find that learning boolean logic acts as a nice initialization for the subsequent reasoning tasks on both ReClor and DREAM. For instance, initializing with $u_{0\sim1}$ improves DeBERTa-V3-base by 3.4% compared to naive fine-tuning on ReClor, and $u_{0\sim1} \rightarrow u_{0\sim2}$ further improves by 4.4%. It is worth noting that u_0 alone does not provide any useful signals (59.0% on ReClor and 80.2% on DREAM), suggesting that it is the boolean logic that we add into the data that enhances the eventual logical performance.

As a contrast, we first train the model on task-specific data and then fine-tune it on boolean logic. We find that more complex logic in ReClor or DREAM does not enable the model to perform any better on $u_{0\sim1}$ or even harms it, confirming our initial idea, that the model may ignore the basic logic during training, even if it appears to handle more complex problems sometimes.

It is the generic form of CLR to pre-learn boolean logic and then learn complex logic.

5.3 Ablation Study

The ablation study is made under negation-only sets. We first discuss the composition of levels to make up the curriculum to perform CLR. We remove some levels from the full curriculum setting $u_{0\sim1} \rightarrow u_{0\sim2} \rightarrow u_{0\sim3} \rightarrow u_{0\sim4}$. Additionally, we include another strong baseline by merging all the

		u_0 *clean%*	$u_{1\sim4}$ *boolean%*	$u_{5\sim8}$
NAI	$u_{0\sim1}, \cdots, u_{0\sim4}$	95.5	95.8	66.5
CLR	$u_{0\sim1} \rightarrow \cdots \rightarrow u_{0\sim4}$	**96.3**	**96.8**	**79.2**
NAI	$u_{0\sim1}, u_{0\sim3}$	95.4	86.6	55.7
CLR	$u_{0\sim1} \rightarrow u_{0\sim3}$	95.8	92.7	66.4
NAI	$u_{0\sim2}, u_{0\sim4}$	95.8	60.1	51.1
CLR	$u_{0\sim2} \rightarrow u_{0\sim4}$	90.8	82.9	55.8
NAI	$u_{0\sim1}, u_2, u_3, u_4$	95.6	95.6	65.6
CLR	$u_{0\sim1}$	**96.2**	64.6	49.0
	$\rightarrow u_2$	95.9	89.9	57.7
	$\rightarrow u_3$	95.3	94.3	64.8
	$\rightarrow u_4$	95.6	**96.5**	**72.2**

Table 5: Ablation study on DeBERTa-V3-base. We omit the notations of $u_{0\sim2}$ and $u_{0\sim3}$ in "$\cdots$".

training sets together, e.g. $u_{0\sim1}, u_{0\sim2}, u_{0\sim3}, u_{0\sim4}$ and performing naive training. The difference is that CLR strategically samples the training data from easy ones to hard ones rather than uniformly. The results are summarized in Table 5. We find that any leap from the full curriculum can result in a notable performance drop, highlighting the importance of a complete and gradual progression of logical learning. Interestingly, we also find that learning from simpler $u_{0\sim1} \rightarrow u_{0\sim3}$ achieves a better outcome compared to harder $u_{0\sim2} \rightarrow u_{0\sim4}$.

Next, we discuss the composition of samples for each level. We remove the simpler samples that belong to the prior level ($u_{0\sim1} \rightarrow u_2 \rightarrow u_3 \rightarrow u_4$) and see whether the model would forget what it has learned before as a result. From Table 5, we find that the removal process gives comparable results on u_0 and $u_{1\sim4}$, However, when it comes to harder $u_{5\sim8}$, it leads to a performance drop of 6%. These findings underscore the importance of reusing simpler samples when stepping forward to the new level, especially when evaluating on harder or even unseen data like $u_{5\sim8}$.

5.4 Fine-tuning Large Language Models

We also evaluate our method on LLMs. However, fine-tuning LLMs requires a huge amount of resources. As a compromise, recent studies propose several efficient fine-tuning methods that only update a small ratio of parameters within LLMs. We experiment on three models, GPT2-1.5b (Brown et al., 2020), OPT-7b (Zhang et al., 2022a), and LLaMA2-7b (Touvron et al., 2023). They both belong to the decoder-only architecture as ChatGPT. We fine-tune GPT2-1.5b with full parameters and fine-tune the 7b models with the low rank adaption method (LoRA) (Hu et al., 2022).

		u_0 *clean%*	$u_{1\sim4}$ *boolean%*	$u_{5\sim8}$
				GPT2-1.5b
NAI	$u_{0\sim1}, \cdots, u_{0\sim4}$	93.8	99.2	65.6
CLR	$u_{0\sim1}$	**95.6**	74.0	52.8
	$\rightarrow u_{0\sim2}$	94.4	84.6	55.8
	$\rightarrow u_{0\sim3}$	94.1	98.6	71.2
	$\rightarrow u_{0\sim4}$	94.3	**99.9** $_{\uparrow 0.7}$	**79.4** $_{\uparrow 13.8}$
				OPT-7b (LoRA)
NAI	$u_{0\sim1}, \cdots, u_{0\sim4}$	94.3	98.0	63.8
CLR	$u_{0\sim1}$	93.3	68.7	54.2
	$\rightarrow u_{0\sim2}$	94.3	78.2	53.0
	$\rightarrow u_{0\sim3}$	94.7	97.9	64.5
	$\rightarrow u_{0\sim4}$	**95.5**	**98.8** $_{\uparrow 0.8}$	**69.4** $_{\uparrow 5.6}$
				LLaMA2-7b (LoRA)
NAI	$u_{0\sim1}, \cdots, u_{0\sim4}$	97.3	99.4	67.9
CLR	$u_{0\sim1}$	96.3	64.3	48.3
	$\rightarrow u_{0\sim2}$	97.6	86.2	51.8
	$\rightarrow u_{0\sim3}$	**97.7**	98.6	67.8
	$\rightarrow u_{0\sim4}$	97.6	**99.9** $_{\uparrow 0.5}$	**75.9** $_{\uparrow 8.0}$

Table 6: Results of LLMs, including the efficient fine-tuning method (LoRA).

From Table 6, we find that CLR works very well on GPT2-1.5b, achieving a boolean accuracy of 79.4% on $u_{5\sim8}$, outperforming naive training by a notable margin of 13.8%. However, larger-scaled OPT-7b does not yield better results as expected. Specifically, it achieves comparable results on simpler $u_{1\sim4}$, while greatly lags behind much smaller GPT2-1.5b on harder $u_{5\sim8}$. We conjecture that parameter efficient fine-tuning might compromise the acquisition of complex reasoning capability, e.g. multi-nested boolean logic, leading to a non-negligible performance drop.

6 Further Discussion

6.1 Chain-of-Thought

We discuss CoT in more detail, and this part is particularly geared to the current LLMs. It has been shown that when being asked to give the procedure, the model can perform the reasoning more precisely. In the prior section, we show that CoT can assist ChatGPT in achieving better performance on BoolKill. We notice that the intermediate thinking procedure exposed by CoT is equally important. A ideal reasoner can not only make the final answer but also reasonable intermediate results.

However, we find that ChatGPT leans to fall into inconsistent deduction when giving the intermediate results, as we illustrate in Table 7.

For $[a]$, we can first obtain that S_3 is false from

the previous statements since S_0 is true. Hence, the deduction of the model up to this step is correct. In the next step, however, the model draws an incorrect conclusion based on the fact that S_3 is false, that is S_4 is true. Indeed, S_4 should be false since it doesn't match S_3, incurring a wrong final answer. Similar cases can be found in $[b]$. These cases indicate that when making longer reasoning, ChatGPT can fall into mistake in some logical step, even though each step is very easy when cutting individually.

$[a]$ *Query*
S0: A crust is a portion of a world. S1: S0 is a false statement. S2: S1 is a false statement. S3: S2 is a false statement. S4: S3 is a true statement. Is S4 true or false? Let's think step by step.
$[a]$ *Answer*
... S3: S2 is a false statement. It contradicts the prior assessment S2 is true, so S3 is false. S4: S3 is a true statement. It contradicts the prior assessment S3 is false, so S4 is true.
$[b]$ *Query*
S0: Humans sometimes eat clouds. S1: S0 is a true statement. S2: S1 is a false statement. S3: S2 is a true statement. S4: S3 is a true statement. S5: S4 is a false statement. Is S5 true or false? Give the procedure.
$[b]$ *Answer*
... S1: S0 is a true statement. (True assuming S0 is true) S2: S1 is a false statement. (False because S1 is true) S3: S2 is a true statement. (True because S2 is false) S4: S3 is a true statement. (False because S3 is true) ... Therefore, S5 is true based on the evaluation.

Table 7: ChatGPT case study. S_4 in $[a]$ should be false.

6.2 True or False

We take a further look at true-or-false questions, a specific and common question type in MRC and logical end tasks. Specifically, we filter out the samples with questions that contain keywords "true" or "false". In ReClor, there are 173 such samples out of the 500 in its development set. The evaluation results on true-or-false questions are shown in Table 8. We find that both DeBERTa models struggle with seemingly simple true-or-false questions, showing lower accuracy compared to the overall performance. However, the models pre-learned with nested boolean logic showcase a significant improvement, achieving 6.4% and 6.3% points of gain respectively.

		True/False	All
DeBERTa-V3-base	*sp*	51.4	58.2
DeBERTa-V3-base	*boolean* $\rightarrow$ *sp*	**57.8**	62.6
DeBERTa-V3-large	*sp*	67.1	71.4
DeBERTa-V3-large	*boolean* $\rightarrow$ *sp*	**73.4**	74.8

Table 8: Results on true-or-false questions in ReClor.

7 Related Work

The study of boolean operations is the fundamental requirement for a series of challenging tasks, e.g. arithmetical reasoning (Ling et al., 2017), commonsense reasoning (Zellers et al., 2019), reading comprehension (Yang et al., 2018), dialogue comprehension (Sun et al., 2019). We concentrate on the multi-nested boolean logic by augmenting the text with boolean statements. Previous studies analyze more general logical reasoning, e.g. RICA (Zhou et al., 2021), RobustLR (Sanyal et al., 2022a), FaiRR (Sanyal et al., 2022b), by logical paraphrase or contrast sets.

Self-supervised learning methods typically generate learnable inputs on top of unlabeled corpora, e.g. by masking (Devlin et al., 2019), insertion (Wu et al., 2022), sentence reordering (Lan et al., 2020), contrastive learning (Gao et al., 2021), while our method is by imposing a series of sentences to the suffix, actually generating learnable logic. We introduce curriculum learning (Bengio et al., 2009), which allows the model to learn step by step to further facilitate self-supervised learning. Curriculum learning is under-discussed in context of language processing (Xu et al., 2020; Lee et al., 2022).

While deep neural networks are capable of handling very complex tasks, in reality they lean to exploit spurious cues (Goodfellow et al., 2015; Madry et al., 2018; Wu et al., 2023a), and can be powerless to very simple perturbations as a consequence. Our work discloses that language models are poorly skilled at basic boolean logic. In parallel, studies show that language models can be easily fooled by some naive patterns within the text, e.g. lexical overlap (McCoy et al., 2019; Wu et al., 2023c), entity boundary (Yang et al., 2023), word order (Zhang et al., 2019).

We also release a challenging benchmark to evaluate boolean logical reasoning. There are a series of work focusing on constructing challenging logic,

e.g. ReClor (Yu et al., 2020), HotpotQA (Yang et al., 2018), ANLI (Nie et al., 2020).

8 Conclusion

This paper provides a quantified analysis on the multi-nested boolean logic. We flag the deficiency in the state-of-the-art language models in terms of such basic capability, which will inevitably cause pitfalls in dealing with more complex reasoning tasks. For this, we propose *Curriculum Logical Reasoning*, a new self-supervised learning method to empower language models with foundational logical capability. We also show that our idea can act as a cornerstone learning method for general logical reasoning.

Limitations

We cannot exhaust all the arrangements of curriculum to perform CLR, which could potentially achieve even better performances. We have discussed the potential risk of chain-of-though as secondary contribution of our work, which will be interesting to study in the future. Our method to introduce nested boolean logic is general, while our experiments are based on one source. Another option is to collect data from more general corpus or specific domains of interest, which is promising. Eventually, we do not have enough resources to run large language models above 7b.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on*

Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mingyu Lee, Jun-Hyung Park, Junho Kim, Kang-Min Kim, and SangKeun Lee. 2022. Efficient pre-training of masked language model via concept-based curriculum masking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 7417–7427. Association for Computational Linguistics.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 158–167. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3428–3448. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4885–4901. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4658–4664. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Soumya Sanyal, Zeyi Liao, and Xiang Ren. 2022a. Robustlr: A diagnostic benchmark for evaluating logical robustness of deductive reasoners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9614–9631. Association for Computational Linguistics.

Soumya Sanyal, Harman Singh, and Xiang Ren. 2022b. Fairr: Faithful and robust deductive reasoning over natural language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin,*

Ireland, May 22-27, 2022, pages 1075–1093. Association for Computational Linguistics.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge dataset and models for dialogue-based reading comprehension. *Trans. Assoc. Comput. Linguistics*, 7:217–231.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Hongqiu Wu, Ruixue Ding, Hai Zhao, Boli Chen, Pengjun Xie, Fei Huang, and Min Zhang. 2022. Forging multiple training objectives for pre-trained language models via meta-learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 6454–6466. Association for Computational Linguistics.

Hongqiu Wu, Ruixue Ding, Hai Zhao, Pengjun Xie, Fei Huang, and Min Zhang. 2023a. Adversarial self-attention for language understanding. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 13727–13735. AAAI Press.

Hongqiu Wu, Yongxiang Liu, Hanwen Shi, Hai Zhao, and Min Zhang. 2023b. Toward adversarial training on contextualized language representation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hongqiu Wu, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. 2023c. Rethinking masked language modeling for chinese spelling correction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10743–10756. Association for Computational Linguistics.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6095–6104. Association for Computational Linguistics.

Yifei Yang, Hongqiu Wu, and Hai Zhao. 2023. Attack named entity recognition by entity boundary interference. *CoRR*, abs/2305.05253.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *CoRR*, abs/2305.10601.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 93–104. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin,

Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1298–1308. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *CoRR*, abs/2210.03493.

Pei Zhou, Rahul Khanna, Seyeon Lee, Bill Yuchen Lin, Daniel Ho, Jay Pujara, and Xiang Ren. 2021. RICA: evaluating robust inference capabilities based on commonsense axioms. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7560–7579. Association for Computational Linguistics.

A BoolKill

	Task	PN ratio	Size	Length	Vocab
BoolKill	binary	1:1	6000	36∼88	14315

Table 9: Statistics of BoolKill.

SciTail	*[Premise]* The planet Mercury is the closest of the planets to the Sun. *[Hypothesis]* Mercury is closest to the sun. *[Label]* Entail
Context-question (k=0)	S0: The planet Mercury is the closest of the planets to the Sun. So, Mercury is closest to the sun. Is S0 true or false? *[Label]* True
BoolKill (k=1)	S0: The planet Mercury is the closest of the planets to the Sun. So, Mercury is closest to the sun. S1 is a false statement. Is S1 true or false? *[Label]* False
BoolKill (k=3)	S0: The planet Mercury is the closest of the planets to the Sun. So, Mercury is closest to the sun. S1 is a false statement. S2 is a true statement. S3 is a false statement. Is S3 true or false? *[Label]* True

Table 10: Illustration of some samples from BoolKill.